今日から気分はプログラマ…？
Microsoft Excelで電源をコントロールしてみよう

かつて、ちょっとした自動測定のシステムを作ろうとすると、相当に難しいイメージがありました。比較的やさしいと言われる言語（BASICなど）であっても、パソコンに不慣れな人には、おいそれと踏み入れない雰囲気があったように思います。

しかし昨今、WindowsやMacintoshのようなGUI（絵を介したインターフェース）を全面的に採用したパソコンの普及とともに、計測の世界も変わりつつあります。BASICやC言語のようなプログラム言語を深く知らなくとも、あらかじめ用意されたパーツを、あたかも玩具のブロックのように組み立てるだけでプログラムが簡単に作れるようになってきました。

そこで、先般発売となりました当社のパワーサプライ・コントローラPIA4800シリーズと、みなさんご存知の表計算ソフト「Microsoft Excel 97」に含まれる「Visual Basic Editor」を使用して、電源をコントロールする簡単な例をご紹介したいと思います。

ここではVisual Basicによる制御のおおまかな理解を目的にしているため、細かい解説は省略してあります。実務に使えるようになるためには、やはり基礎知識の学習が必要になることは、お含みいただきたいと思います。

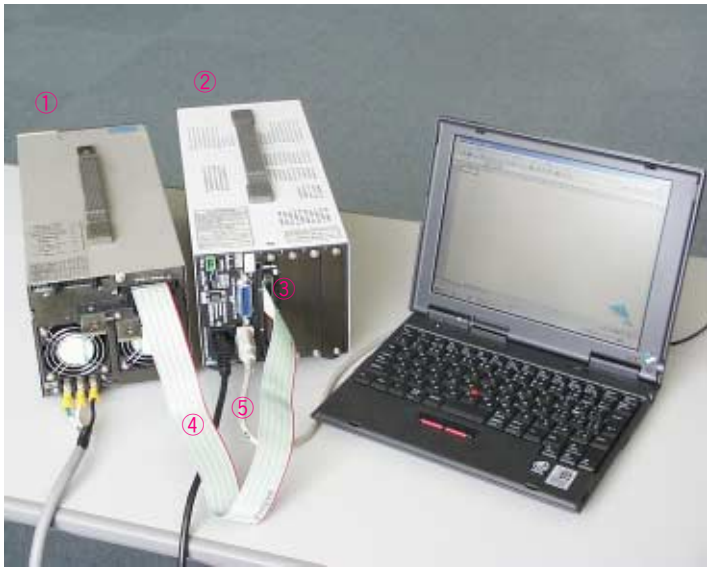
VB／VBAについて

Visual Basic（VB）とは、Microsoft社が開発したWindows上で動く汎用的なプログラミング言語です。そのVBにExcel、Word、Access等のMicrosoft Office特有の機能を加味して、より複雑な処理を記述できる言語が、Visual Basic for Applications（VBA）と呼ばれるものです。ここではExcel97に内蔵されているVBAプログラムを作るための統合開発環境であるVisual Basic Editorを使って電源を制御します。なお、VBAは本家VBに対して若干の違いがありますが、殆どの部分は同じです。また、統合開発環境とは、プログラムの編集、フォームデザイン、プログラムの実行デバックなどを一つの画面で視覚的に行えることを指します。

準備

今回は、パワーサプライ・コントローラPIA4810とそのコントロールボードOP01-PIAを使用し、コンピュータから1台の直流電源PAK60-12Aの出力電圧を制御（定電圧制御）してみます。

まず取扱説明書に従いPIA4810、OP01-PIA及びIF01-PAK-Aを内蔵したPAK60-12Aを接続します<写真1>。具体的には、OP01-PIAをPIA4810後面パネルのSLOT 1に装着し、PAK60-12AをそのOP01のCH1側に26pinケーブルで接続します。またPAKをフルリモートで運転するので、取説に従ってPAK本体後面のディップスイッチを正しくセットします。またコンピュータとPIA4800の接続はRS-232Cで接続します（このケーブルはクロスケーブルです）。



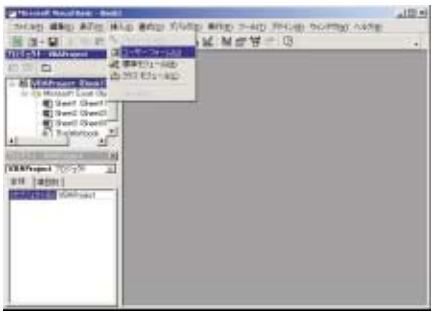
<写真1>
今回の記事に使用した機器。なお、接続状態をわかりやすくするために、各機器とも背面パネルを手前にして撮影しています。
①直流電源 PAK60-12A
②パワーサプライ・コントローラ PIA4810
③コントロールボード OP01-PIA
④26pinケーブル
⑤RS-232Cケーブル

Visual Basic Editorを起動

その前に、PIA4810または4830に付属するVisual Basicドライバがコンピュータにインストールされている必要がありますが、ここでは既にインストールされている前提で話を進めます。またPIA4800シリーズでは「コンフィグレーション」と呼ばれる作業を行なって、接続されている電源機器を事前にPIA4810／4830に認識させておく必要がありますが、この作業も既に終えているものとします。

まず、Excel97を起動します。そして[ツール]メニューの中から[マクロ]－[VisualBasic Editor]を選ぶとVisual Basic Editorの画面が現れます<図1>。そして[挿入]メニュー－[ユーザーフォーム]コマンドで、[ツールボックス]の[コントロール]と、[User Form]を表示します<図2>。

次に[ツール]メニュー－[その他のコントロール]－[利用可能なコントロール]から[PIA4800 DRIVER]または、[PIA4800 Class]にチェックマークを入れ、[OK]ボタンを押します。すると[ツールボックス]に[PIA]コントロールのアイコンが表示されますので、[PIA]コントロールをドラッグして



<図1>
Visual Basic Editorの画面

[User Form]に貼り込みます。

このコントロールのNameプロパティはデフォルトではPia48001ですが、これでは長いので単にpiaと変更します。また、コマンドボタンも同じく[UserForm]に貼り込みします。これでプログラム記述の準備が整いました<図3>。

プログラムを記述する

まず、UserForm上のCommandButton1をダブルクリックします。するとこのボタンを押した時に実行されるコード(ボタンハンドラと言います)が表示されます。

```
Private Sub CommandButton1_Click()

End Sub
```

このブロックの内側にコードを書きます。まず、機器ドライバを使用したプログラムで最初にしなければならないのは、機器との接続です。これはConnectメソッドで行ないます。このメソッドは他の機種のドライバでも同様です。

```
pia.Connect "COM1"
```

この一行で機器との接続を開始します。"COM1"はRS-232Cポートの1番を意味します。GPIBでアドレス2を使用する場合は"DEV2"となります。数値部分は別のアドレス番号と差し替えてもかまいません。

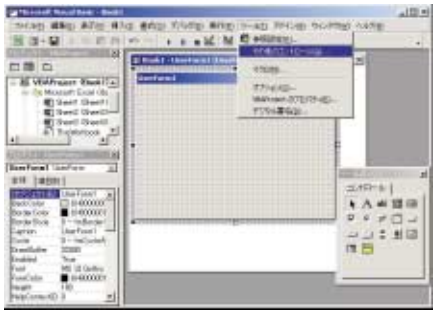
この例では、PIA4810のSLOT1にOP01-PIAを装着しました。このSLOTに装着されたカードの事をPIA4800機器ドライバではModuleと呼びます。またこのModuleのCH1にPAK60-12Aを接続しました。ここで以下のコードを書きます。

```
Dim pak As PIA4800Lib.Supply
Set pak = pia.Modules(1).Supplies(1)
```

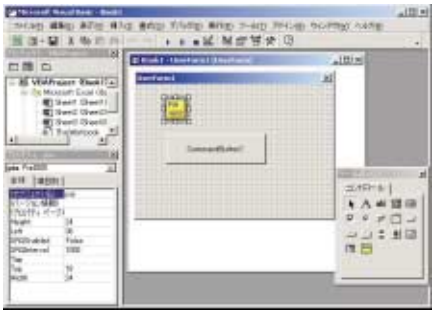
一見ちょっとわかりにくいかもしれませんが、これはSupply型(電源型)のオブジェクト変数pakを宣言し、pia(これはPIA4810全体を指します)からModule1、CH1の要素を抽出している記述なのです。ModuleやSupplyが英語の複数形で記述されているのは、VBやVBAで「コレクション(集合体)」と呼ばれている機能を利用しているためです。先の文を説明すると、「piaのModulesコレクションから1番目を引き出し、その中にあるSuppliesコレクションから1番目を引き出す」となります。まさにSLOT1に装着したOP01-PIAのCH1に接続されたPAK60-12Aということになります。番号を変えれば他のスロットやコネクタに接続された電源装置も簡単に扱えます。あとは、単に電圧・電流・出力SWを設定するだけです。

```
pak.Voltage = 10
pak.Current = 1
pak.Output = True
```

この様に電源装置を扱うオブジェクトにはVoltage, Current等のプロパティが用意されています。実際にはもっと数多くの機能がプロパティ・メソッドを通じて用意されています。これらの詳細はVB/VBAから参照可能なオンラインリファレンスで見ることができます。



<図2>
User Formを表示



<図3>
PIAコントロールとコマンドボタンを
UserFormに貼り込んだところ

プログラムを実行する

以上でプログラムは完成です。[実行]メニュー―[sub/ユーザフォームの実行]コマンドを選択すると実行画面になります。そして「CommandButton1」ボタンをクリックすると、先のプログラムで記述した10V／1Aの設定になりOUTPUTもON状態になります。別の電圧値などにしたい場合は、単にVoltageプロパティに与える数値を変えるだけです。

エラー処理について

GPIBやRS-232Cを使って電源や計測器を制御するプログラムにとって忘れてならないのが通信エラーです。ケーブルが抜けてしまっていたり装置の電源が落ちていたりした場合に備えたプログラムを書くのは面倒です。PIA4800シリーズの機器ドライバでは、通信エラーが発生した場合の処理も簡単になっています。というのは、このドライバでは通信エラーをVB/VBAの実行時エラーとして扱うからです。記述方法はいたって単純、ON ERROR GOTO 文を書くだけです。

```
  ON ERROR GOTO Instr_Exp:
  pia.Connect "COM1:
    ...
Instr_Exp:
  MsgBox Err.Description
```

こう記述すれば、通信エラーが発生した際にエラー処理ルーチンへジャンプし、エラーが発生した原因を表示することができます。

このように、機器ドライバを使用したプログラムは、従来のプログラムの書き方と比較して飛躍的にコード記述量を減らすことができ、アプリケーション固有の内容に専念する事ができます。

当社ホームページにはPIA4800シリーズ以外の他機種用にも同様の機器ドライバが用意されています。ぜひご使用いただければと思います。

近藤　亮
(菊水電子工業株式会社　ソフト技術グループ)